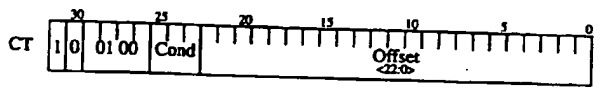
	Document ID	U	Title	Current OR
1	US 20020 13871 5 A1	☐	Microprocessor executing data transfer between memory and register and data transfer between registers in response to single push/pop instruction	712/225
2	US 62126 30 B1	☐	Microprocessor for overlapping stack frame allocation with saving of subroutine data into stack area	712/242
3	US 55749 28 A	☐	Mixed integer/floating point processor core for a superscalar microprocessor with a plurality of operand buses for transferring operand segments	712/23
4	US 55420 60 A	☐	Data processor including a decoding unit for decomposing a multifunctional data transfer instruction into a plurality of control codes	712/208
5	US 55242 11 A	☐	System for employing select, pause, and identification registers to control communication among plural processors	709/220
6	US 55176 64 A	☐	RISC system with instructions which include register area and displacement portions for accessing data stored in registers during processing	712/41
7	US 54936 87 A	☐	RISC microprocessor architecture implementing multiple typed register sets	712/23
8	US 54427 61 A	☐	Method by which packet handler inserts data load instructions in instruction sequence fetched by instruction fetch unit	712/205
9	US 53275 66 A	☐	Stage saving and restoring hardware mechanism	710/260
10	US 52416 79 A	☐	Data processor for executing data saving and restoration register and data saving stack with corresponding stack storage for each register	710/260
11	US 51094 95 A	☐	Method and apparatus using a source operand list and a source operand pointer queue between the execution unit and the instruction decoding and operand processing units of a pipelined data processor	712/207
12	US 50310 96 A	☐	Method and apparatus for compressing the execution time of an instruction stream executing in a pipelined processor	711/169
13	US 48336 40 A	☐	Register bank change including register to register transfer in a data processing system	718/108
14	US 43386 63 A	☐	Calling instructions for a data processing system	712/228

VCJSR — Conditional Jump to Subroutine

Format



Assembler Syntax

VCJSR.cond #Offset

where cond = {un, lt, eq, le, gt, ne, ge, ov}.

Description

Jump to subroutine if Cond is true. This is not a delayed branch.

If Cond is true, VPC + 4 (the return address) is saved onto the return address stack. If not, the execution continues with VPC + 4.

Operation

```
If ( (Cond == VCSR[SO,GT,EQ,LT]) | (Cond == un) ) {
    if (VSP<4> > 15) {
        VISRC<RASO> = 1 ;
        signal ARM7 with RASO exception ;
        VP_STATE = VP_IDLE ;
    } else {
        RSTACK[VSP<3:0>] = VPC + 4 ;
        VSP<4:0> = VSP<4:0> + 1 ;
        VPC = VPC + sex(Offset<22:0> * 4) ;
    }
} else VPC = VPC + 4;
```

Exception

Return address stack overflow.